基于折射原理的混合型花朵授粉算法 *

崔丽群，张 晨，郑宝林，周四宏

(辽宁工程技术大学 软件学院，辽宁 葫芦岛 125105)

摘 要：针对花朵授粉算法收敛速度慢，寻优精度低的缺陷，提出基于折射原理的混合型花朵授粉算法（refrHFPA）。算法首先利用和声搜索算法提升算法收敛速度，然后利用折射原理提高种群的多样性，帮助算法跳出局部最优，提升寻优精度。实验利用 8 个测试函数，对比其他群智能算法，结果表明 refrHFPA 算法在收敛速度和寻优精度方面均有显著的提高。

关键词：花朵授粉算法；和声搜索算法；折射原理；种群多样性

中图分类号：TP301.6 doi: 10.3969/j.issn.1001-3695.2017.11.0731

Hybrid flower pollination algorithm based on refraction principle

Cui Liqun, Zhang Chen, Zheng Baolin, Zhou Sihong

(*School of software Liaoning Technical University*, *Huludao Liaoning* 125105, *China*)

Abstract: This paper proposed a hybrid flower pollination algorithm (refrHFPA) based on refraction principle for the slow convergence rate of flower pollination algorithm and low optimization accuracy. The algorithm firstly used the harmony search algorithm to improve the convergence speed of the algorithm, then it used the refraction principle to improve the diversity of the population, and helped the algorithm to jump out of the local optimal and improved the accuracy of optimization. It used eight test functions to compare other intelligent algorithms, and the results show that the refrHFPA algorithm has a significant improvement in convergence speed and optimization accuracy.

Key words: flower pollination algorithm; harmony search algorithm; refraction principle; population diversity

0 引言

花朵授粉算法（flower pollination algorithm，FPA）[1]是一种新型元启发式智能算法，由英国学者 Yang 于 2012 年首次提出。由于 FPA 算法具有参数较少、实现简单、易与其他算法进行组合优化，且转换概率 p 可以较好地调节全局搜索与局部搜索之间平衡等优点，目前已在函数优化领域得到了应用。例如，中国学者 Zhou 等人[2]应用改进的花朵授粉算法优化无人海底车辆的路径规划；贺圣彦等人[3]将该算法应用到 PID 参数优化；Abdel-Baset 等人[4]利用该算法进行比例优化。但是花朵授粉算法也具有仿生算法的典型缺点，如易陷入局部极值、后期收敛速度慢等问题。为了解决这些问题，文献[5]将 FPA 与萤火虫算法相结合，在局部搜索部分引入自适应因子并自适应调整转换概率 p。文献[6]利用高斯变异对全局搜索进行扰动增强种群的多样性，将 Powell 法引入局部搜索，以提升其局部搜索能力。文献[7]将差分进化的变异、交叉和选择操作应用到花朵授粉算法帮助算法跳出局部最优，以增强全局搜索能力。FPA 算法因提出时间短，迫切需要理论研究及应用领域的扩展。虽然以上文献对 FPA 算法的寻优能力有一定程度上的改进，但仍存在寻优精度低、后期收敛速度慢等问题。

针对花朵授粉算法速度和精度的问题，本文将折射原理与和声搜索算法引入到花朵授粉算法中，提出基于折射原理的混合型花朵授粉算法（refrHFPA）。该算法将和声搜索算法引入到 FPA 算法中，以提高原算法的寻优速度；利用折射原理提高原种群多样性，使 FPA 算法跳出局部最优，以提高算法的收敛精度。

1 基础理论

1.1 花朵授粉算法

花朵授粉算法是一种模拟开花植物授粉进程的智能算法。授粉方式分为非生物自花授粉和生物异花授粉。非生物自花授粉是指花粉的传播不需要生物作为传播者，而是通过风进行传播，这种传播方式范围较小，被视为局部搜索。生物异花授粉是两种开花植物进行授粉，授粉过程是由蜜蜂等传粉者进行传播，该过程因可在大范围内进行传播，被视为全局搜索。全局搜索与局部搜索之间的转换通过转换概率 p 来进行调节。现实

收稿日期：2017-11-01；**修回日期：**2017-12-27 **基金项目：**国家自然科学基金资助项目（61172144）；辽宁省教育厅资助项目（L2012113）

作者简介：崔丽群（1969-），女，副教授，主要研究方向为图像与视觉信息计算（chen17627@qq.com）；张晨（1994-），女，硕士，主要研究方向为智能信息处理；郑宝林（1996-），男，主要研究方向为软件工程；周四宏（1995-），男，主要研究方向为软件工程.

生活中的每一棵显花植物都可以产生数百万甚至更多的花粉配子。为了简化问题，通常假设每棵显花植物只开一朵花，而且每朵花只产生一个花粉配子，这代表每个花粉配子只对应一个解。

根据开花植物的授粉过程，花朵授粉算法具有如下四条规律[1]：

a)生物异花授粉是携带花粉的传粉者进行全局搜索，方式符合莱维飞行；

b)非生物自花授粉被视为局部搜索过程；

c)花的恒常性被认为是繁殖概率，这种概率与所涉及的两朵花的相似性成正比；

d）转换概率 $p\in[0,1]$，受物理距离和其他因素的影响，在整个授粉活动中，p 值的选取非常重要。

1.2　折射原理

光的折射是指光从一种介质斜射入另一种介质，它是自然界中存在的一种普遍的自然现象[8]，如图 1 所示。其中 α_1 为入射角，α_2 为折射角，O 为交界点。光的传播方向在不同介质的交界处发生偏折。

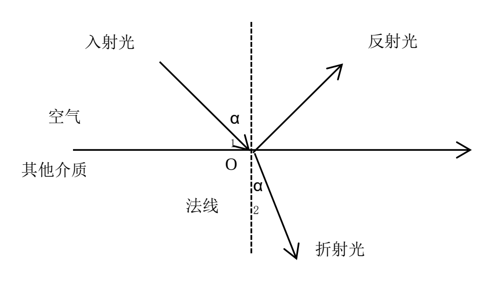


图 1　光的折射原理

斯涅耳通过实验确立了折射定律，即介质的绝对折射率 n 为入射角和折射角的正弦比[9]，公式为

$$n=\sin\alpha_1/\sin\alpha_2 \tag{1}$$

2　改进的花朵授粉算法

2.1　基于和声搜索算法的花朵授粉算法

和声搜索算法（Harmony search，HS）是由韩国学者 Geem 等人[10]于 2001 年提出的启发式优化算法。由文献[10]可知，HS 算法由于随机选择音调的作用使其全局搜索能力较强。因此将 HS 算法引入 FPA 算法中，得到基于和声搜索算法的花朵授粉算法（HSFPA）。HSFPA 算法通过将 HS 算法的最优解作为初始解来提高算法的寻优精度。和声搜索算法具体步骤如下：

a)对 HMS（初始解个数）、HMCR（和声记忆库考虑概率）、PAR（和声微调概率）、BW（音调微调带宽）等参数进行设置。

b)算法进行初始化，随机生成 HMS 个和声，形成和声记忆库。和声记忆库为：

$$HM=\left[\begin{array}{cccc|c} x_1^{\,1} & x_2^{\,1} & \cdots & x_n^{\,1} & f(x^1)\\ x_1^{\,2} & x_2^{\,2} & \vdots & x_n^{\,2} & f(x^2)\\ \vdots & \vdots & \cdots & \vdots & \vdots\\ x_1^{\,HMS} & x_2^{\,HMS} & \cdots & x_n^{\,HMS} & f(x^{HMS}) \end{array}\right]$$

c)生成一个新的和声。新和声的每一个音调通过如下三种机理产生：

（a）学习和声记忆库。

（b）若新解第一个变量以 HMCR 的概率选自和声记忆库，则按照公式（2）对音调进行微调，反之则在变量范围内（和声记忆库外）随机选择一个值作为新解。

$$x_1'=\begin{cases} x'\pm r_1\times BW_1 & 若 r_2<PAR\\ x_1 & 其他 \end{cases} \tag{2}$$

其中：r_1、r_2 为符合[0,1]均匀分布的随机数。

（c)随机选择音调。

（d)将生成的新和声和原有和声记忆库中的最差值进行比较，若优于最差值则对和声记忆库进行更新，反之则不变。

（e）判断是否符合迭代结束条件，符合则结束迭代，不符合则重复（c）和（d）直到符合条件。

2.2　基于折射原理的花朵授粉算法

本文将折射原理引入 FPA 算法，以增加种群的多样性，

帮助算法跳出局部最优，提高算法的收敛速度和寻优精度。具体过程如图 2 所示。将空气与其他介质的临界处看为 X 轴，X 是搜索区域的上下界，取值范围为[a，b]，点 X_{best} 是当前迭代的最优解，X_1 为 X_{best} 正上方的入射光点，Y 为点 X_1 的折射点，Y_1 则为点 Y 在交界处的投影即点 X_{best} 的折射点。将 OX_1 的距离设为 H，OY 的距离设为 H_1，交界点 O 为中心点（a+b）/2。

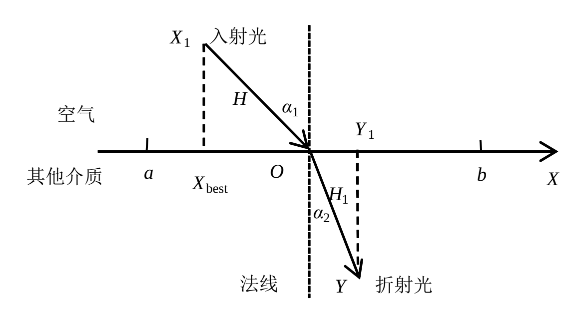


图 2　折射原理优化 FPA 算法原理示意

由图 2 可知：

$$\sin\alpha_1=((a+b)/2-X_{best})/H \tag{3}$$

$$\sin\alpha_2=(Y_1-(a+b)/2)/H_1 \tag{4}$$

由式（3）（4）可知,折射率 n 可由式（5）表示。

$$n=\sin\alpha_1/\sin\alpha_2=\frac{((a+b)/2-X_{best})/H}{(Y_1-(a+b)/2)/H_1} \tag{5}$$

设 H/H_1=f，可将式（5）改为

$$Y_1=(a+b)/2+((a+b)/2-X_{best})/(fn) \tag{6}$$

算法每进行一次迭代，X_{best} 都通过式（6）进行折射，得到折射解 Y_1，通过判断 X_{best} 与 Y_1 的适应度值大小，对当前的全局最优解进行更新。

2.3　基于折射原理的混合型花朵授粉算法

将 HS 算法的最优解作为 FPA 算法的初始解，可在一定程度上提升算法的收敛精度，但是 HS 算法的随机性较大，有时

会使算法陷入局部最优。因此利用折射原理增强种群的多样性，以利于算法跳出局部最优，提高算法的寻优精度和收敛速度。改进后的算法流程如图 3 所示。具体实现步骤如下：

a)初始化算法的各参数。包括 p，HMCR,PAR 等参数。

b)求出每个解的适应度值。

c)利用 HS 算法求出当前最优解作为初始值。

d)当 p>rand（rand 为[0,1]均匀分布的随机数）时，按照式（7）对解进行更新，按照式（8）对步长进行更新；反之则按照式（9）对解进行更新，最后进行越界处理。

$$X_i^{t+1} = X_i^t + L(X_i^t - g_*) \tag{7}$$

其中：X_i^{t+1}、X_i^t 分别是指向量 X_i 迭代 t+1 次和 t 次后的解；g_* 代表当前最优解。

$$L \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}},(s >> s_0 > 0) \tag{8}$$

其中：L 代表步长；Γ（λ）是标准的伽玛函数，λ=1.5。

$$X_i^{t+1} = X_i^t + \varepsilon(X_j^t - X_g^t) \tag{9}$$

其中：X_j^t 和 X_k^t 代表来自相同植物不同花的花粉；ε 是[0,1]均匀分布的随机数。

e)求得当前最优解和最优解的适应度值。

f)对当前最优解按照式（6）进行处理，得到折射点 Y_1，然后对该点的函数值进行判断。若优于当前最优解，则更新全局最优解和最优解的适应度值，反之不变。

g)判断是否满足实验设置的最大迭代次数或固定收敛精度，若满足则输出结果结束算法，否则转至 c)。

3　仿真实验及结果分析

为了检验 refrHFPA 算法的有效性，利用八个测试函数在 MATLAB 的环境下进行测试，并将其测试结果与 FPA、HSFPA、refrFPA 算法进行比较。其中，八个测试函数如下：

（1）$f_1(x) = \sum_{i=1}^{n} x_i^2$，$x_i \in [-5.12, 5.12]$，i=1,2,3...,n,该函数的理论最优值为 0。

（2）$f_2(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$，$x_i \in [-100,100]$，i=1,2,3...,n,该函数的理论最优值为 0。

（3）$f_3(x) = \sum_{i=1}^{n} |x_i \sin(x_i) + 0.1x_i|$，$x_i \in [-10,10]$，该函数的理论最优值为 0。

（4）$f_4(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 + 1 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}})$，$x_i \in [-600,600]$，i=1,2,3...n,该函数的理论最优值为 0。

（5）$f_5(x) = \sum_{i=1}^{n} \left| x_i - \frac{1}{i} \right|$，$x_i \in [-10,10]$，i=1,2,3...,n,该函数的理论最优值为 0。

（6）$f_6(x) = 20 + e - 20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))$，$x_i \in [-32,32]$，i=1,2,3...,n,该函数的理论最优值为 0。

（7）$f_7(x) = \prod_{i=1}^{n}|x_i| + \sum_{i=1}^{n}|x_i|$，$x_i \in [-10,10]$，i=1,2,3...,n,该函数的理论最优值为 0。

（8）$f_8(x) = \sum_{i=1}^{n} (10^6)^{\frac{i-1}{n-1}} x_i^2$，$x_i \in [-100,100]$，i=1,2,3...,n,该函数的理论最优值为 0。

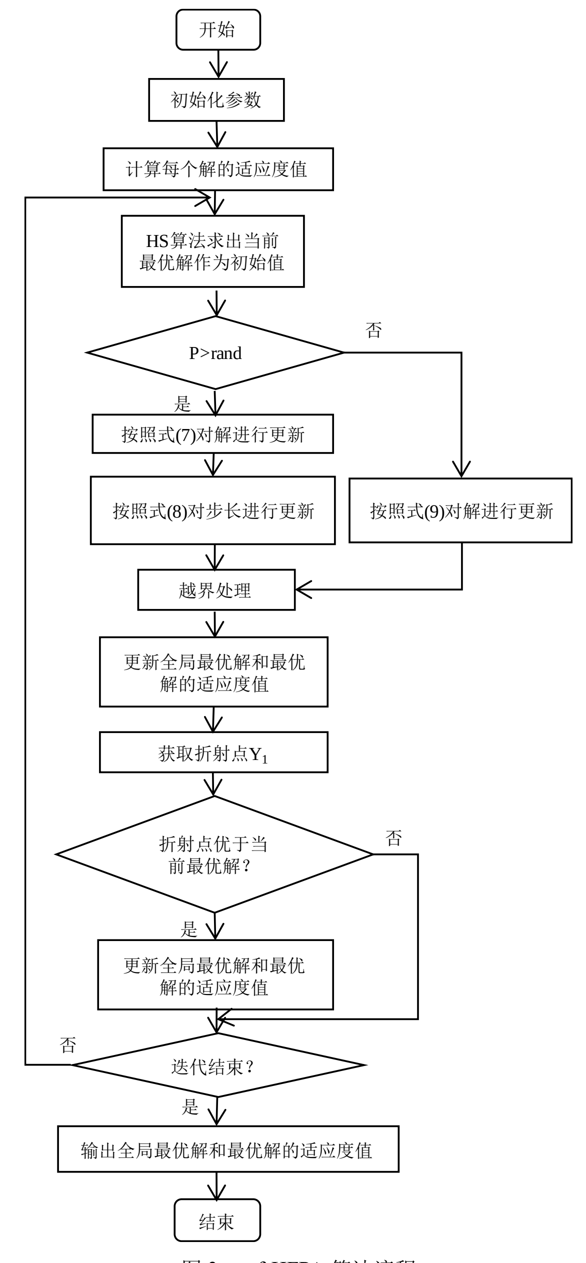


图 3　refrHFPA 算法流程

3.1　参数 f 的分析与选取

参数 f 的取值直接影响折射解在 X 轴上的位置，取 f 在预估范围[0,1]内，对单峰函数 f_1 和 f_2、多峰函数 f_4 和 f_6 独立运行 20 次。判断 f 取何值时可以有效帮助算法跳出局部最优。本实验设置种群数为 20，最大迭代次数为 300，refrHFPA 算法参数中 p=0.2[11]，λ=1.5，HMCR=0.95，PAR=0.3，BW=0.01，n=2[8]。实验结果如表 1 所示。

表 1　f 取不同值时 refrHFPA 算法的性能分析

函数	维数	f	最优值	平均值	最差值
f_1	30	0.2	2.492e-01	5.4612e-01	9.545e-01
		0.4	1.675e-01	7.1101e-01	1.2157
		0.6	**3.5e-03**	**2.3523e-02**	**1.315e-01**
		0.8	**6.1e-04**	**3.7303e-03**	**8.6e-03**
		1	**6.62e-04**	**1.6072e-03**	**3.1e-03**
f_2	30	0.2	1.6512e+02	2.7669e+02	3.6554e+02
		0.4	1.0628e+02	3.0286e+02	7.2461e+02
		0.6	**8.3564**	**1.5407e+01**	**2.4545e+01**
		0.8	**3.0175**	**8.2552**	**4.5821e+01**
		1	**2.8153**	**4.9436**	**7.2475**
f_4	30	0.2	2.4283	3.82006	5.0583
		0.4	1.7756	3.71579	6.7429
		0.6	**1.0333**	**1.06872**	**1.1426**
		0.8	**3.353e-01**	**8.3804e-01**	**1.0272**
		1	**2.949e-01**	**7.7102e-01**	**1.0487**
f_6	30	0.2	5.2427	6.94923	1.0569e+01
		0.4	4.4971	7.02611	1.1209e+01
		0.6	**8.233e-01**	**2.49892**	**4.2241**
		0.8	**3.862e-01**	**9.3248e-01**	**1.6789**
		1	**1.898e-01**	**5.7248e-01**	**1.253**

由表 1 可以看出，参数 f 取值 [0.6 ,1]相比[0, 0.4]寻优结果有 1~3 个数量级的提高。为了进一步确定参数 f 取值对算法寻优结果的影响，实验采用 f_1 与 f_4，收敛情况如图 4 所示。

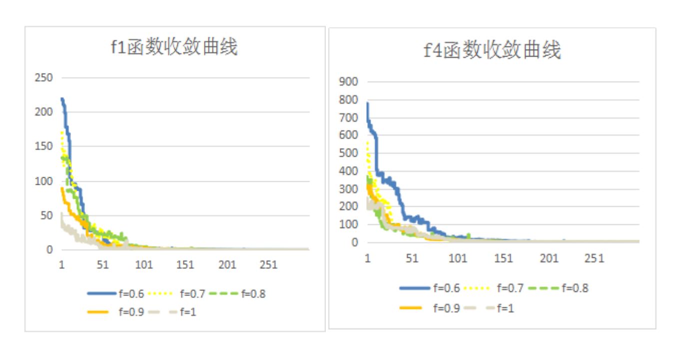


图 4　不同 f 取值测试函数的收敛曲线

由图 4 可见，f 值在[0.8,1]内时收敛速度更快，因此将 f 按照式（10）进行取值，初始值设为 0.8。

$$f = 0.8 + 0.2 \times rand() \tag{10}$$

3.2　测试方案

本文共设计五种仿真实验方案：第一种是固定种群数量及迭代次数，通过八种测试函数测试 FPA、HSFPA、refrFPA 及 refrHFPA 算法在低维环境下的寻优能力；第二种是设定一个高维的实验环境，判断 refrHFPA 算法在高维环境下是否可以得到较好的收敛精度；第三种是固定收敛精度，对比这四种算法的最小收敛次数、平均收敛次数和收敛率；第四种是 refrHFPA 算法在不同维度下，判断其是否具有较低的时间复杂度；第五种是将 refrHFPA 算法与近两年改进后的 FPA 算法进行对比，验证本文算法的可行性。

本文设置的实验参数如下：

FPA 算法中转换概率 p=0.8[1]，λ=1.5。

HSFPA 算法中 p=0.8[1]，λ=1.5，HMCR=0.9，PAR=0.3，BW=0.1。

refrFPA 算法中 p=0.2[11]，λ=1.5，n=2[8]，f=0.8，。

refrHFPA 算法参数设置与小节 3.1 相同。

3.2.1 固定迭代次数性能分析

1）低维下的性能分析

本文实验方案设置种群数为 20，迭代次数为 1 000， f_3 维数为 10，其余函数维数为 30。每种算法对每个测试函数均独立运行 50 次 ，计算其最优值、最差值、平均值和寻优成功率。其中寻优成功的标准是实际最优值与理论最优值的差值小于 0.001。实验结果如表 2 所示。

表 2　refrHFPA 算法在低维上的性能分析

函数	算法	平均值	最差值	最优值	成功率
f_1	FPA	2.74751	4.483	1.5283	0
	HSFPA	0.06378	0.13458	0.01626	0
	refrFPA	1.8486e-08	4.4378e-08	3.426e-09	100%
	refrHFPA	**9.4645e-12**	**1.6008e-10**	**7.5141e-15**	**100%**
f_2	FPA	1.2752e+03	3.3109e+03	1.0687e+02	0
	HSFPA	2.9799e+01	4.4754e+01	1.1784e+01	0
	refrFPA	5.696e-04	1.1301e-03	2.1231e-04	92.8%
	refrHFPA	**2.7042e-05**	**1.3952e-04**	**2.5498e-06**	**100%**
f_3	FPA	1.48352	2.429	5.389e-01	0
	HSFPA	5.8429e-03	3.06e-02	1.9e-03	0
	refrFPA	9.2467e-01	2.5223	6.5e-03	0
	refrHFPA	**9.1811e-09**	**8.6994e-08**	**3.3573e-10**	**100%**
f_4	FPA	1.4417e+01	1.1886e+02	4.77	0
	HSFPA	1.37221	1.9232	1.1005	0
	refrFPA	5.0578e-03	4.34e-02	4.6823e-06	50%
	refrHFPA	**1.4265e-10**	**5.89e-09**	**5.9064e-14**	**100%**
f_5	FPA	1.3112e+01	1.8652e+01	9.1526	0
	HSFPA	1.01672	1.4035	0.6329	0
	refrFPA	2.3643e-03	3.5e-03	1.8e-03	0
	refrHFPA	**5.1908e-04**	**9.3451e-04**	**2.6926e-04**	**100%**
f_6	FPA	5.81155	7.9429	2.8798	0
	HSFPA	2.95162	4.4069	2.036	0
	refrFPA	7.8168e-02	7.838e-01	8.5053e-04	7%
	refrHFPA	**2.4028e-05**	**4.0876e-05**	**1.8607e-05**	**100%**
f_7	FPA	1.4951e+1	2.2939e+1	9.4265	0
	HSFPA	1.58679	2.6311	4.712e-01	0
	refrFPA	7.8514e-05	1.6539e-04	2.9294e-05	100%
	refrHFPA	**1.248e-07**	**1.5261e-06**	**4.3813e-09**	**100%**
f_8	FPA	1.0406e+05	2.8598e+05	3.392e+04	0
	HSFPA	4.7374e+04	7.2367e+04	1.9956e+04	0
	refrFPA	1.7043e-02	5.08e-02	1.3e-03	0
	refrHFPA	**2.6904e-07**	**2.9884e-06**	**9.7789e-11**	**100%**

从表 2 可以看出，针对本文选取的测试函数，refrHFPA 算法的最优值、最差值及平均值均优于 FPA、HSFPA 和 refrFPA 算法；除 f_6 外，HSFPA 较于 FPA 算法，寻优结果均有 1~2 个数量级的提高；除 f_3 外，对于不同的测试函数 refrFPA 算法较于原算法均有 4~8 个数量级的提高。此外，refrHFPA 算法对于

这八种测试函数的寻优率达到 100%，并且最优值、最差值和平均值也远高于其他三种算法。为了更直观地比较 refrHFPA 算法与其他三种算法寻优能力，本文对于八个测试函数的收敛情况如图 5 所示。

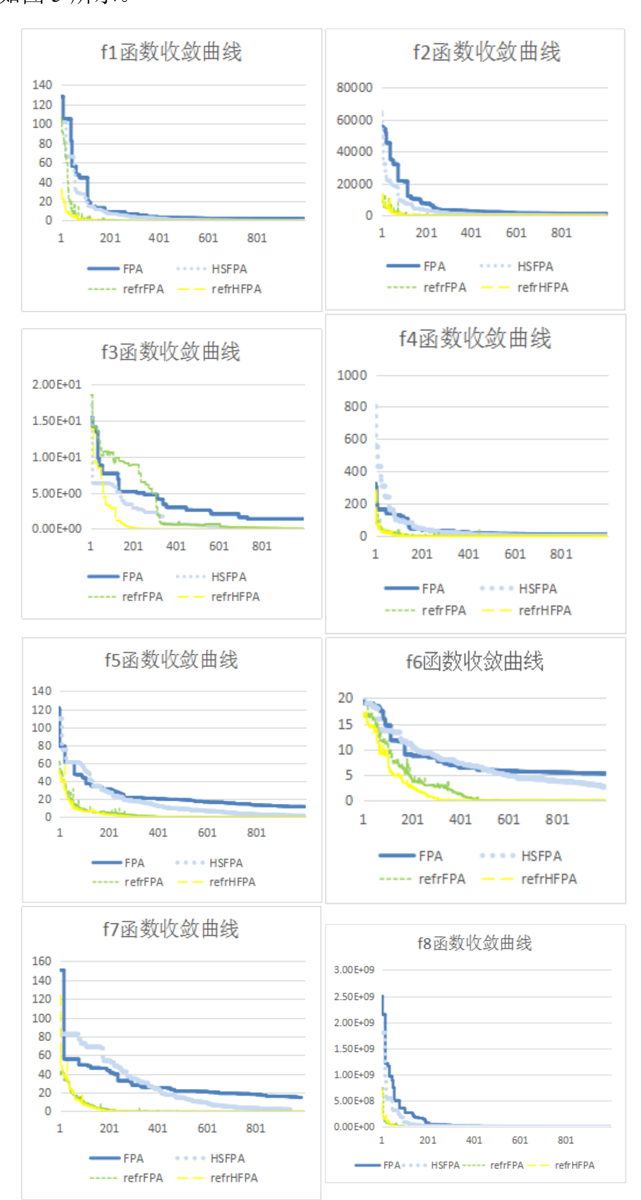


图 5　四种算法对于测试函数的收敛曲线

从图 5 可以看出，HSFPA 算法在对测试函数 f_5-f_7 进行测试时陷入局部最优，但 refrHFPA 算法却能很好地跳出局部最优，说明引入折射定理可以有效的增加种群多样性，帮助算法跳出局部最优，提升算法的收敛速度和寻优精度。

2）高维下的性能分析

目前处理的很多数据多为高维数据，如多媒体图形图像和视频数据等，本文算法在高维环境下进行实验。维数设为 100，迭代次数设为 4 000，其余参数不变。对于每个测试函数，将原花朵授粉算法与本文提出的花朵授粉算法均独立实验 50 次，若实际最优值与理论最优值的差值小于 0.001 时代表寻优成功。实验结果如表 3 所示。

表 3　refrHFPA 算法在高维上的性能分析

函数	维数	算法	平均值	最差值	最优值	成功率
f_2	100	FPA	5.4706e+03	9.8813e+03	2.8358e+03	0
		refrHFPA	**1.5171e-07**	**3.586e-07**	**4.5143e-08**	**100%**
f_4	100	FPA	5.1105e+01	8.8088e+01	3.1268e+01	0
		refrHFPA	**2.5373e-13**	**3.6189e-12**	**3.3307e-16**	**100%**
f_6	100	FPA	5.6286	9.3972	2.8912	0
		refrHFPA	**1.8174e-05**	**1.8273e-05**	**1.8172e-05**	**100%**
f_8	100	FPA	4.7518e+07	8.7437e+07	1.9636e+07	0
		refrHFPA	**5.6762e-06**	**1.9884e-04**	**7.2084e-12**	**100%**

表 3 表明，在高维环境下 refrHFPA 与 FPA 算法相比，在寻优精度上有 5~15 个数量级的提高。因此 refrHFPA 算法可以跳出局部最优，不易随着维数的增加而陷入“维灾难”。

3.2.2 固定收敛精度性能分析

收敛率是寻优成功的次数占实验总次数的百分比。为了验证 refrHFPA 算法的收敛性，在固定精度的前提下，对每个测试函数独立运行 50 次，将其最小收敛代数、平均收敛代数、收敛率与 FPA、HSFPA、refrFPA 算法相比较。实验中四种算法的种群数均为 20，维数设定为 10，最大迭代次数为 2 500。实验结果如表 4 所示。

表 4　四种算法在固定收敛精度下的性能分析

函数	维数	固定精度	算法	最小收敛代数	平均收敛代数	收敛率
f_1	10	1.0e-05	FPA	961	1225.58	100%
			HSFPA	582	685.97	100%
			refrFPA	224	290.43	100%
			refrHFPA	**161**	**216.18**	**100%**
f_2	10	1.0e-07	FPA	1760	2059.17	82%
			HSFPA	1076	1182.36	100%
			refrFPA	633	666.14	100%
			refrHFPA	**558**	**601.08**	**100%**
f_3	10	1.0e-03	FPA	fail	fail	0
			HSFPA	927	1193.71	100%
			refrFPA	fail	fail	0
			refrHFPA	**325**	**397.36**	**100%**
f_4	10	1.0e-07	FPA	fail	fail	0
			HSFPA	fail	fail	0
			refrFPA	fail	fail	0
			refrHFPA	**202**	**594.38**	**100%**
f_5	10	1.0e-03	FPA	1076	1165.2	100%
			HSFPA	872	959.79	100%
			refrFPA	437	455.43	100%
			refrHFPA	**304**	**356.72**	**100%**
f_6	10	1.0e-03	FPA	1542	2056.3	86%
			HSFPA	910	1045.07	100%
			refrFPA	500	730.29	100%
			refrHFPA	**240**	**441.2**	**100%**
f_7	10	1.0e-03	FPA	1856	2115.23	62%
			HSFPA	844	929.43	100%
			refrFPA	258	344.36	100%
			refrHFPA	**209**	**294.96**	**100%**
f_8	10	1.0e-07	FPA	2271	2479.1	20%
			HSFPA	1486	1611.21	100%
			refrFPA	465	606	100%
			refrHFPA	**450**	**582.6**	**100%**

从表 4 可以看出，对于这八个测试函数，HSFPA、refrFPA 与 refrHFPA 算法无论是最小收敛代数还是平均收敛代数均优于原算法。由此可见 HS 算法与折射定理的引进对于 FPA 算法均是有效的。此外，refrHFPA 算法的最小收敛代数和平均收敛代数在四种算法中最小，且收敛率均为 100%。因此 refrHFPA 算法的收敛性要优于其他三种算法，并具有有效性。

3.2.3 refrHFPA 算法的时间复杂度分析

本文将 refrHFPA 与 FPA 算法在相同环境下，通过运行时间来观察本文算法是否满足时间复杂度较低的要求。实验中种群数量设置为 10，迭代次数为 300，维数分别设为 10、100、200，每种算法对于每种测试函数均独立运行 20 次。refrHFPA 与 FPA 算法在不同维度下的最小运行时间与平均运行时间如表 5 所示。

表 5　refrHFPA 算法时间复杂度分析

函数	维数	算法	最优值	平均值
f_1	10	FPA	0.302	0.31624
		refrHFPA	**0.902**	**0.94136**
	100	FPA	0.603	0.62993
		refrHFPA	**1.662**	**1.70079**
	200	FPA	0.929	0.966
		refrHFPA	**2.453**	**2.53729**
f_4	10	FPA	0.339	0.3765
		refrHFPA	**0.903**	**1.30135**
	100	FPA	0.703	0.74143
		refrHFPA	**1.662**	**2.7271**
	200	FPA	1.081	1.12068
		refrHFPA	**2.453**	**4.18715**
f_6	10	FPA	0.34	0.35907
		refrHFPA	**1.238**	**1.35307**
	100	FPA	0.698	0.73064
		refrHFPA	**2.458**	**2.60107**
	200	FPA	1.079	1.15293
		refrHFPA	**3.838**	**4.02456**
f_7	10	FPA	0.311	0.32585
		refrHFPA	**1.041**	**1.15021**
	100	FPA	0.652	0.68729
		refrHFPA	**2.15**	**2.41129**
	200	FPA	1.005	1.0425
		refrHFPA	**3.245**	**3.4529**

从表 5 可以看出，refrHFPA 与 FPA 算法相比运行时间略长，但 refrHFPA 算法无论在低维还是高维环境下的寻优精度都远高于 FPA 算法，而且 refrHFPA 算法的最小运行时间与平均运行时间仍然较短，所以可以看出 refrHFPA 算法是可行的。

3.2.4 改进智能算法比较实验

本文将 refrHFPA 算法与文献[6,7]相比较，测试函数为文献中的部分函数。本实验方案的参数设置与参考文献相同，GMPFPA 和 DEFPA 算法的实验数据均来自于参考文献，结果如表 6 和 7 所示。

表 6　refrHFPA 算法与 GMPFPA 算法优化性能对比

函数	维数	算法	平均值	最差值	最优值
f_1	30	GMPFPA	1.4323e-18	1.5429e-17	1.0003e-30
		refrHSFPA	**7.5053e-30**	**4.2212e-29**	**3.4197e-33**
f_2	30	GMPFPA	1.1166e-13	1.3006e-12	7.886e-13
		refrHSFPA	**2.7426e-17**	**1.6592e-16**	**2.2627e-18**
f_6	30	GMPFPA	1.3881e-10	1.467e-09	4.4409e-15
		refrHSFPA	**5.7731e-15**	**1.0658e-14**	**3.5527e-15**
f_7	30	GMPFPA	1.5011e-16	2.3621e-15	2.7323e-21
		refrHSFPA	**6.0552e-19**	**2.7656e-18**	**2.2322e-20**

表 7　refrHFPA 算法与 DEFPA 算法优化性能对比

函数	维数	算法	平均值	最差值	最优值
f_1	30	DEFPA	2.6347e-17	1.0206e-16	7.4022e-18
		refrHSFPA	**1.8415e-32**	**6.587e-31**	**1.0809e-34**
f_4	30	DEFPA	5.0093e-15	1.4322e-14	9.992e-16
		refrHSFPA	**0**	**0**	**0**
f_6	30	DEFPA	3.359e-08	6.9112e-08	1.4435e-08
		refrHSFPA	**7.9935e-15**	**2.1316e-14**	**3.5527e-15**
f_7	30	DEFPA	9.2161e-31	2.2095e-30	3.66e-31
		refrHSFPA	**4.6329e-19**	**1.8917e-18**	**1.5876e-20**

由表 6 可以看出，refrHFPA 算法相较于 GMPFPA 算法，单峰函数有 12 个数量级的提高，多峰函数有 3~5 个数量级的提高。由表 7 可以看出，除 f_7 外，refrHFPA 算法较于 DEFPA 算法，在寻优结果上均有 6~16 个数量级的提高。因此，在实验条件相同的情况下，无论是单峰函数还是多峰函数，refrHFPA 算法寻优结果优于参考文献[6,7]算法。

4　结束语

花朵授粉算法作为新型群智能算法，也与其他智能算法一样存在收敛速度慢、寻优精度低的缺点。针对这些不足，本文将 HS 算法引入花朵授粉算法，将其最优解作为算法迭代的初始解，以提高初始解的质量。利用折射定理对当前最优解进行折射，以提升算法的全局搜索能力，提高寻优精度。实验结果表明，本文算法与原算法和其他改进算法相比，收敛速度较快、最优解质量较高，具有一定的可行性和有效性。以后，将在花朵授粉算法的参数值设置、推广算法应用领域等方面做进一步研究。

参考文献：

[1] Yang Xinshe. Flower pollination algorithm for global optimization [C]// Proc of International Conference on Unconventional Computation and

Natural Computation. 2012: 240-249.

[2] Zhou Yongquan, Wang Rui. An improved flower pollination algorithm for optimal unmanned undersea vehicle path planning problem [J]. International Journal of Pattern Recognition and Artificial Intelligence, 2016, 30 (4): 1659010.

[3] 贺圣彦，曹中清，余胜威. 基于花授粉算法的 PID 参数优化 [J]. 计算机工程与应用, 2016, 52 (17): 59-62.

[4] Abdel-Baset M, Hezam I M. An improved flower pollination algorithm for ratios optimization problems [M]// Applied Mathematics and Information Sciences Letters. 2015: 83-91.

[5] 卞京红，贺兴时，杨新社. 基于萤火虫算法的自适应花授粉优化算法 [J]. 计算机工程与应用, 2016, 52 (21): 162-167.

[6] 肖辉辉，万常选，段艳明，等. 融合高斯变异和 Powell 法的花朵授粉优化算法 [J]. 计算机科学与探索, 2017, 11 (3): 478-490.

[7] 肖辉辉，万常选，段艳明. 一种改进的新型元启发式花朵授粉算法 [J]. 计算机应用研究, 2016, 33 (1): 126-131.

[8] 邵鹏，吴志健，周炫余，等. 基于折射原理反向学习模型的改进粒子群算法 [J]. 电子学报, 2015, 43 (11): 2137-2144.

[9] Griffiths, David J. Introduction to Electrodynamics [M]. [S. l.] : PrentIce Hall, 1998. 386-389.

[10] Geem Z W, Kim J H, Loganathan G V. A new heuristic optimization algorithm: harmony search [J]. Simulation, 2001, 76 (2): 60−68.

[11] Draa A. On the performances of the flower pollination algorithm-qualitative and quantitative analyses [J]. Elsevier Science Publishers B. V. , 2015, 34 (C): 349-371.

[12] 曲良东，何登旭，黄勇. 自适应改进和声——单纯形进化算法研究 [J]. 计算机应用研究, 2013, 30 (3): 676-678.

[13] Zhao S Z, Suganthan P N, Pan Q K, *et al.* Dynamic multi-swarm particle swarm optimizer with harmony search [J]. Expert Systems with Applications, 2011, 38 (4): 3735-3742.

[14] He X, Yang Xinshe, Karamanoglu M, *et al.* Global convergence analysis of the flower pollination algorithm: a discrete-time markov chain approach [J]. Procedia Computer Science, 2017, 108: 1354-1363.

[15] Zaem D, Mezioud C, Draa A. On the efficiency of the binary flower pollination algorithm: application on the antenna positioning problem [J]. Applied Soft Computing, 2016, 47 (C): 395-414.